**FAS EXAM RESULT Mobile App**

Heshan Karunaratne

**University of Sri Jayewardenepura, Sri Lanka**

**B.Sc. (Honors) in App Sci 2018 April 2018**

**FAS EXAM RESULT Mobile App**

A thesis

Submitted for the Industrial based Project(ASC 408 10.0) to

The Department of Computer Sciences of the

University of Sri Jayewardenepura

in partial fulfillment of the

requirement for the

Bachelor of Science (Honors) Degree in Applied Sciences

In the field of Computer Science

By

Heshan Karunaratne

University of Sri Jayewardenepura, Sri Lanka

December 2018

**DECLARATION**

The work described in this thesis was carried out by me in collaboration with the Department of Computer Sciences, University of Sri Jayewardenepura, Sri Lanka under the guidance of Prof. R. G. N. Meegama and has not been submitted elsewhere.

---------------------------- 31th January 2017

Heshan Karunaratne Date of submission

----------------------------

Prof. R. G. N. Meegama

Internal supervisor

(University of Sri Jayewardenepura)

-----------------------------

Mr. D.D.A.Gamini

Head/ Department of Computer Sciences

University of Sri Jayewardenepura

-------------------------------

Prof. Sudantha Liyanage

Dean/ Faculty of Applied Sciences

University of Sri Jayewardenepura

**ACKNOWLEDGEMENT**

I take this opportunity to express my gratitude to all who supported me to accomplish the research project. I would like to express my appreciation towards Prof. R.G.N.Meegama, Senior Lecturer, Department of Computer Sciences, Faculty of Applied Sciences, University of Sri Jayewardenepura for his guidance, supervision and lending necessary resources for the completion of this project. Also my sincere gratitude goes towards the all the senior lecturers in the Department of Computer Sciences and specially Dr. Ravindra De Silva for supporting with the project.

Furthermore I would also like to offer my sincere thanks to Dr.Pahan Godakubura, Senior Lecturer and Coordinator of B.Sc (Hons) Degree in Applied Science, Faculty of Applied Sciences, University of Sri Jayewardenepura for guiding us during the course. Last but not least, I am grateful to my parents and friends at University of Sri Jayewardenepura for their constant help and feedback during the project in various ways.

**LIST OF ABBREVIATIONS**

ADT - Android Development Tool

AVD - Android Virtual Device

IDE - Integrated Development Environment.

SDK - Software Development Kit

API - Application Programming Interface

APK – Android Application Package

XML - Extensible Markup Language

AOSP -Android Open Source Project

REST - Representational State Transfer

URI - Uniform Resource Identifiers

NPM – Node Package Manager

GPA – Grade Point Average

**Table of Contents**

Table of Figures

Abstract

Chapter 1- Introduction

* + 1. Overview

1.2 Problem definition

1.3 Objectives

1.4 Proposed Solution

Chapter 2 - Technologies Adopted

2.1 Android

2.1.1 Android updates

2.1.2 Android Manifest file

2.1.3 Android Studio

2.1.4 Android Virtual Device

2.2 Wamp Server

2.2.1 MySQL Database

2.3 Node JS

2.4 Postman

2.5 Restful Web Services

Chapter 3 - Methodology

3.1 Setting up the Development Environment

3.1.1 Java

3.1.2 Android studio

3.1.3 Node JS

3.2 Preparing the Database

3.3 Developing the API

3.4 Developing the Android Application

3.5 Retrofit

### 3.5.1 Retrofit Instance

### 3.5.2 Interface

### 3.5.3 Model Class

Chapter 4 – Results and Discussion

4.1 Results

4.2 Discussion

Chapter 5 - Conclusion and Future Works

5.1 Conclusion

5.2 Future Works

References

**Table of Figures**

Figure 1: Performance & Evaluation Service

Figure 2: Error Page of PES

Figure 3: Proposed System

Figure 4: Manifest File

Figure 5: Wamp Server

Figure 6: Postman

Figure 7: JDK Accept License agreement to download

Figure 8: Click the relevant supporting version for OS

Figure 9: JDK Install process initiation

Figure 10: Continue to install

Figure 11: Selecting location to install JDK

Figure 12: Change Environment variable

Figure 13: Edit the Java Home variable

Figure 14: Android Studio

Figure 15: Android install initial process

Figure 16: Choose Components

Figure 17: Choosing a location to install

Figure 18: Start a new Project

Figure 19: Node js downloading website

Figure 20: Node js install initiation

Figure 21: Install necessary packages

Figure 22: Check node js version from terminal

Figure 23: MySQL database and table structure

Figure 24: Application Programming Interface

Figure 25: Android activities

Figure 26: Base URL instance

Figure 27: Retrofit Client with GSON converter

Figure 28: API endpoints

Figure 29: Model class

Figure 30: Splash screen

Figure 31: Login Activity

Figure 32: Home Activity

Figure 33: Result activity after displaying results

Figure 34: GPA Activity

Figure 35: Rank Activity

Abstract

Chapter 1 - Introduction

**1.1Overview**

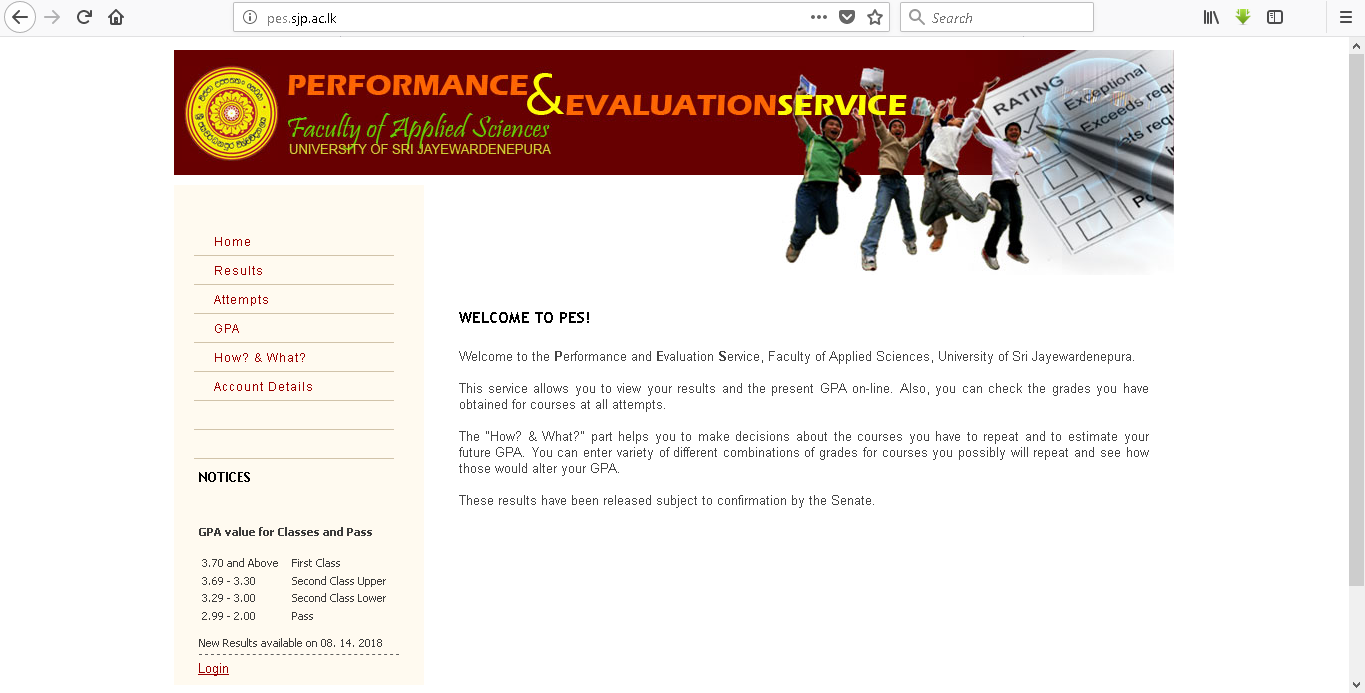
A mobile application is a computer program that is designed to run on a [mobile device](https://en.wikipedia.org/wiki/Mobile_device) such as a [phone](https://en.wikipedia.org/wiki/Smartphone)/[tablet](https://en.wikipedia.org/wiki/Tablet_computer). Mobile applications stand in contrast to [desktop applications](https://en.wikipedia.org/wiki/Desktop_application) which run on [desktop computers](https://en.wikipedia.org/wiki/Desktop_computer), and with [web applications](https://en.wikipedia.org/wiki/Web_application) which run in [mobile web browsers](https://en.wikipedia.org/wiki/Mobile_web_browser). Each day thousands of mobile apps are published to the Google Play and to the App Store. Some of these mobile apps are games, social networks and e-commerce apps. Each app is different and the methodologies are always evolving, but this is the standard process when developing mobile apps. Mobile app development process typically includes idea, strategy, design, development and deployment phases.

Google Play Store (previously Android Market) is a [digital distribution](https://en.wikipedia.org/wiki/Digital_distribution) service operated and developed by [Google](https://en.wikipedia.org/wiki/Google). It serves as the official [app store](https://en.wikipedia.org/wiki/App_store) for the [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) operating system, allowing users to browse and download applications developed with the [Android software development kit (SDK)](https://en.wikipedia.org/wiki/Android_SDK) and published through Google Play Store. As of 2017, Google Play features over 3.5 million Android applications. Developers in over 150 countries can distribute apps on Google Play, though not every location supports merchant registration.

To distribute apps, developers have to pay a one-time $25 registration fee for a Google Play Developer Console account. Developers receive 70% of the application price, while the remaining 30% goes to the distribution partner and operating fees. App developers can control which countries an app is distributed to, as well as the pricing for the app and in-app purchases in each country.

**1.2Problem Definition**

Students of the Faculty of Applied Sciences of University of Sri Jayewardenepura have been using the web portal to check their exam results.

 Figure 1: Performance & Evaluation Service

When they want to check results they must use a PC or a laptop. Some students may not have a laptop to check results. They cannot check results from anywhere without a laptop and internet connection. Sometimes the website fails to load when lot of students try to log at the same time so because of these reasons a proper way to check results is needed.

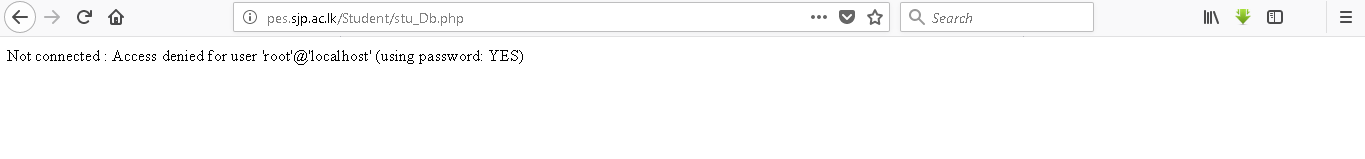


Figure 2: Error Page of PES

**1.3 Objectives**

The main aim of this project was to develop a mobile based application for the students of Faculty of Applied Sciences, University of Sri Jayewardenepura to check exam results easily and efficiently. Accordingly, the following objectives will be achieved upon completing the project.

* To check exam results based on year, subject and both
* To check No of attempts based on year, subject and both
* To get Subject GPA and Overall GPA
* To get the rank from each subject

**1.4 Proposed Solution**

Proposed system is an android based user friendly application that will enable the students of Faculty of Applied Sciences to check results and get to know their GPA. Added feature of this application is that students can get to know their ranks based on the subjects so that they can find what their position in that subject stream.

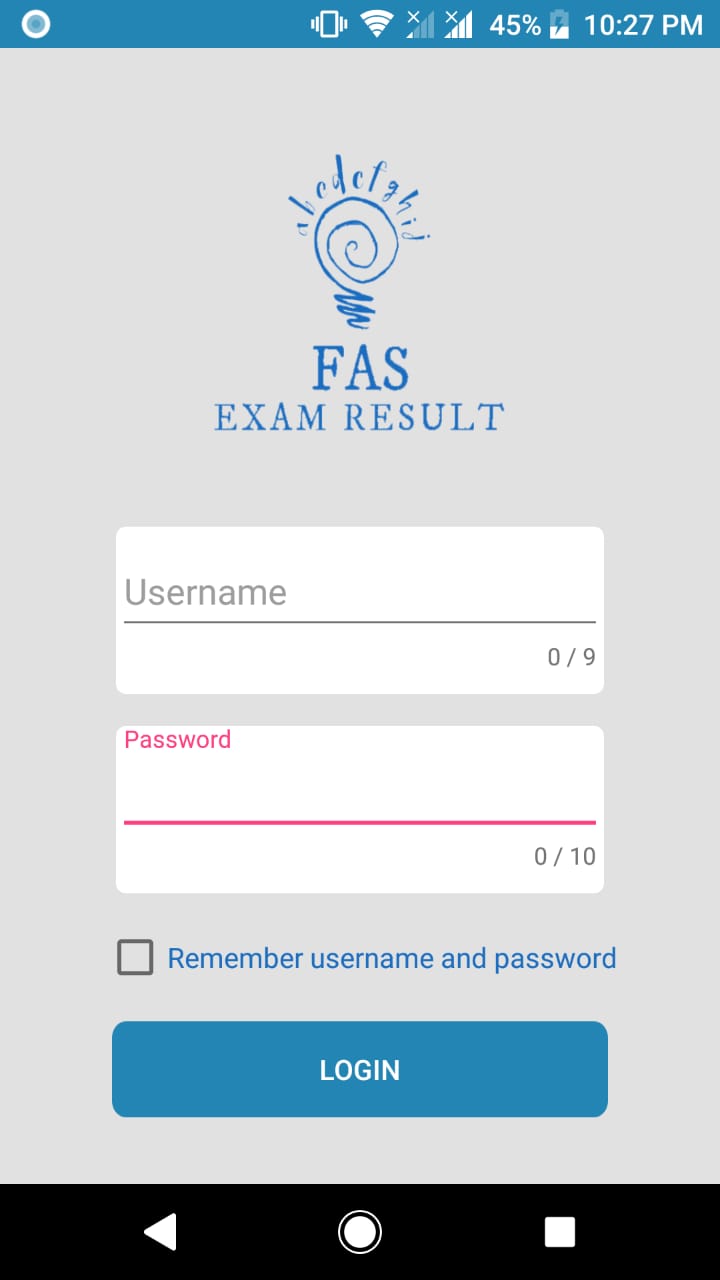


Figure 3: Proposed System

Chapter 2 - Technologies Adopted

**2.1 Android**

Android is a software platform and operating system for mobile based on the Linux kernel which is currently developed by Google. It provides a rich application framework that allows building innovative apps for mobile devices in a java language environment and for developers, Android innovation leads to build powerful yet differentiated applications that use the latest mobile technologies. Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers and also android designed specialized user interfaces for televisions (Android TV), cars (Android Auto) and wrist watches (Android ware).Android operating system uses virtual keyboard and touch inputs that loosely correspond to real-world actions like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects.

Android apps are built as a combination of distinct components that can be invoked individually. An individual activity provides a single screen for a user interface, and a service independently performs work in the background. From one component, can start another component or even a component in a different app using an intent. Android provides an adaptive app framework which allows providing unique resources for different device configurations. The operating system has gone through multiple major releases, with the current version being [9.0 "Pie"](https://en.wikipedia.org/wiki/Android_Pie), released in August 2018. The core Android source code is known as Android Open Source Project and it is primarily licensed under the [Apache License](https://en.wikipedia.org/wiki/Apache_License).[1]

**2.1.1 Android updates**

Google is launching all the android versions with a numerical code and name that’s so far been themed after sweets and desserts, in the alphabetical order. The Android versions are[2],

* Android Beta 1.x
* Android Cupcake 1.5
* Android Donut 1.6
* Android Éclair 2.1
* Android Froyo 2.2
* Android Gingerbread 2.3
* Android Honeycomb 3.2
* Android Ice Cream Sandwich 4.0
* Android Jelly Bean 4.1
* Android Jelly Bean 4.2
* Android Jelly Bean 4.3
* Android Kit Kat 4.4
* Android Lollipop 5.0
* Android Marshmallow 6.0
* Android Nougat 7.0
* Android Oreo 8.0
* Android Pie 9.0

**2.1.2 Android Manifest file**

Every app project must have an AndroidManifest.xml file at the root of the [project source set](https://developer.android.com/studio/build/index.html#sourcesets). Among other things, the manifest file is required to declare the following. The Android build tools use app’s package name to determine the location of code entities when building the project. The components of the app, which include all activities, services, broadcast receivers, and content providers are included in the manifest file. The permissions that the app needs in order to access protected parts of the system and also declares any permissions that other apps must have if they want to access content from the app.[3]

Structure of the Manifest file

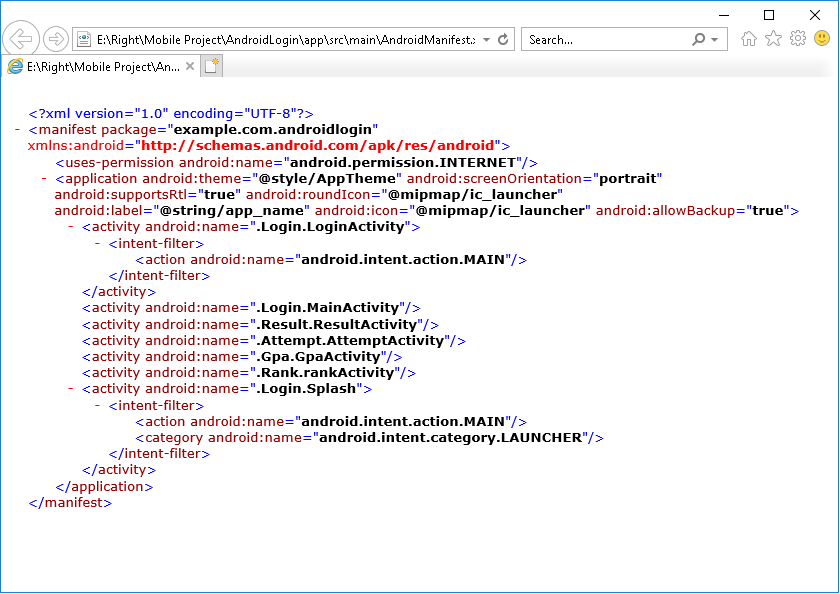


Figure 4:Manifest File

**2.1.3 Android Studio**

Android Studio is the IDE for Android that was announced in 16th of May 2013 at the Google I/O developers’ event, and is intended as an alternative to Eclipse. The first stable build was released in December 2014, starting from version 1.0 and the most recent version being 3.1.4 by the time of writing. Android Studio is based on the Java IDE called IntelliJ. All IntelliJ products share the same shell IDE. Most of the tooling in Android Studio is very similar to Eclipse, such as shortcuts, designers, and code editors. Following features are provided in the current stable version of Android Studio:

* [Gradle](https://en.wikipedia.org/wiki/Gradle) build in support.
* Android [refactoring](https://en.wikipedia.org/wiki/Code_refactoring) and quick fixes.
* [Lint](https://en.wikipedia.org/wiki/Lint_(software))ing tools to increase performance, usability and version compatibility.
* Template based wizards to create common Android designs and components.
* Drag and drop UI components, option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG) on multiple screen configurations.
* Android Virtual Device to run and debug apps in the Android studio.

**2.1.4 Android Virtual Device**

An Android emulator is a virtual Android device running on the computer. The Android emulator mimics all of the hardware and software features of a typical mobile device, except that it cannot place an actual phone call. The emulator allows an application developer to test an Android application on different API levels without using a physical device. An Android Virtual Device is a device configuration that is run within the Android emulator. It works with the emulator to provide a virtual device-specific environment in which to install and run Android apps. The AVD Manager provides a graphical user interface in which a developer can model different configurations of Android devices, which are required by the Android emulator.

**2.2 Wamp Server**

The acronym WAMP refers to a set of free (open source) applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySQL handles the database components, while PHP, Python, or PERL represents the dynamic scripting languages. For the project MYSQL Database was handled by WAMP Server.

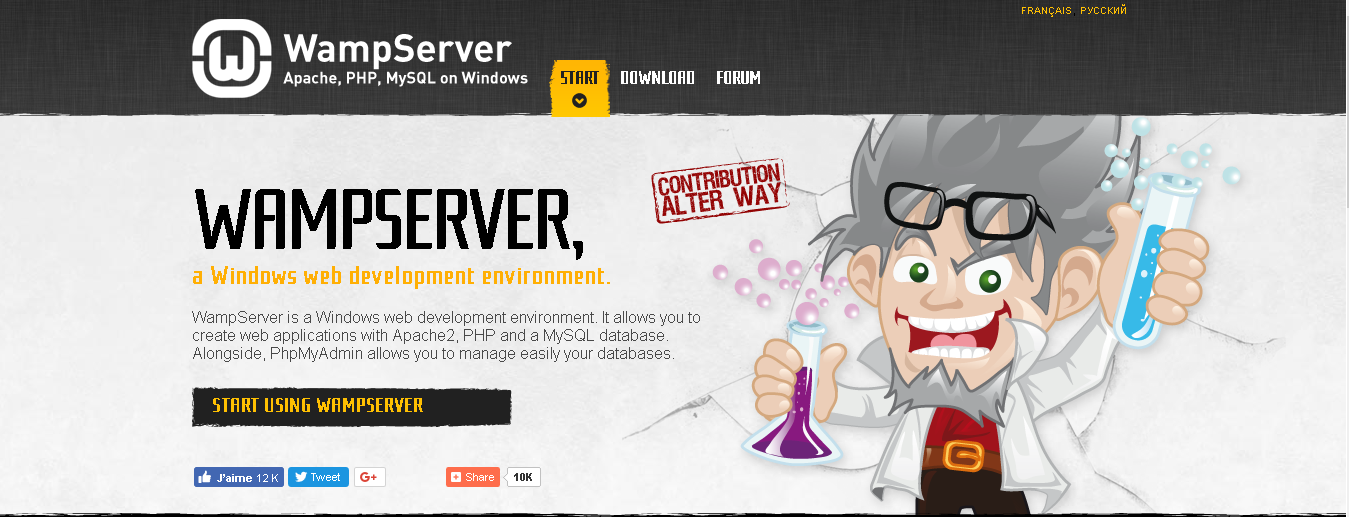


Figure 5:Wamp Server

**2.2.1 MySQL Database**

MySQL is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system). The MySQL development project has made its [source code](https://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL is owned by [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). MySQL is based on a [client-server](https://searchnetworking.techtarget.com/definition/client-server) model. The core of MySQL is MySQL server, which handles all of the database instructions. MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded into separate applications. MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer. MySQL is a central component of the [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) open-source web application software stack. LAMP is an acronym for "[Linux](https://en.wikipedia.org/wiki/Linux), [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL and [Perl](https://en.wikipedia.org/wiki/Perl)/[PHP](https://en.wikipedia.org/wiki/PHP)/[Python](https://en.wikipedia.org/wiki/Python_(programming_language))". MySQL is also used in many high-profile, large-scale [websites](https://en.wikipedia.org/wiki/Website), including [Google](https://en.wikipedia.org/wiki/Google), [Facebook](https://en.wikipedia.org/wiki/Facebook), [Twitter](https://en.wikipedia.org/wiki/Twitter), [Flickr](https://en.wikipedia.org/wiki/Flickr) and [YouTube](https://en.wikipedia.org/wiki/YouTube).

**2.3 Node JS**

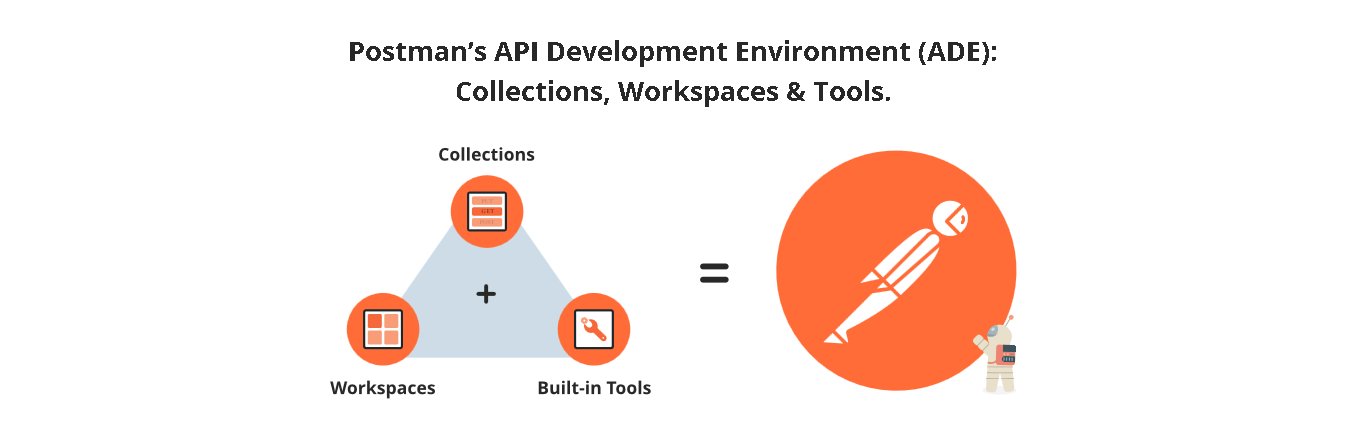
Node.js is an [open-source](https://en.wikipedia.org/wiki/Open-source_software), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [JavaScript](https://en.wikipedia.org/wiki/JavaScript) [run-time environment](https://en.wikipedia.org/wiki/Runtime_system) that executes JavaScript code outside of a browser. JavaScript was used primarily for [client-side scripting](https://en.wikipedia.org/wiki/Client-side_scripting), in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript to write Command Line tools and for [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) running scripts server-side to produce [dynamic web page](https://en.wikipedia.org/wiki/Dynamic_web_page) content before the page is sent to the user's web browser. Node.js represents a "JavaScript everywhere" paradigm, unifying [web application](https://en.wikipedia.org/wiki/Web_application) development around a single programming language, rather than different languages for server side and client side scripts.

Though .js is the conventional [filename extension](https://en.wikipedia.org/wiki/Filename_extension) for JavaScript code, the name "Node.js" does not refer to a particular file in this context and is merely the name of the product. Node.js has an [event-driven architecture](https://en.wikipedia.org/wiki/Event-driven_architecture) capable of [asynchronous I/O](https://en.wikipedia.org/wiki/Asynchronous_I/O). These design choices aim to optimize [throughput](https://en.wikipedia.org/wiki/Throughput) and [scalability](https://en.wikipedia.org/wiki/Scalability) in web applications with many input/output operations, as well as for [real-time Web](https://en.wikipedia.org/wiki/Real-time_Web).[4]

**2.4 Postman**

Postman helps you be more efficient while working with APIs. The need for it arose while one of the developers was creating an API for his project. After looking around for a number of tools, nothing felt just right. The primary features added were a history of sent requests and collections. A number of other features have been added since the initial release. Here is a small list.

* HTTP requests with file upload support
* Formatted API responses for JSON and XML
* Request history
* Basic Auth and OAuth 1.0 helpers
* Keyboard shortcuts to maximize your productivity
* Use collections to organize requests.
* Download and share collections with your team of developers.



**Figure 6: Postman**

**2.5 Restful Web Services**

RESTful web services are built to work best on the Web. Representational State Transfer is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web. The resources are acted upon by using a set of simple, well-defined operations. The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol. RESTful applications are simple, lightweight and fast:

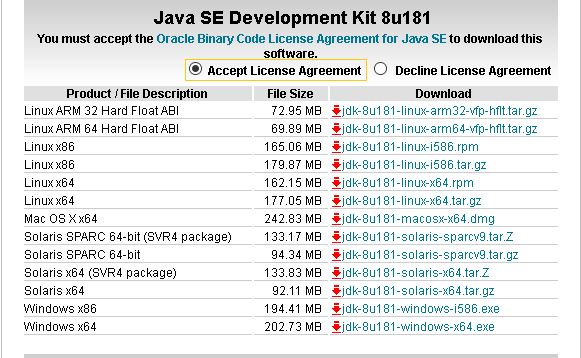
Chapter 3 - Methodology

**3.1 Setting up the Development Environment**

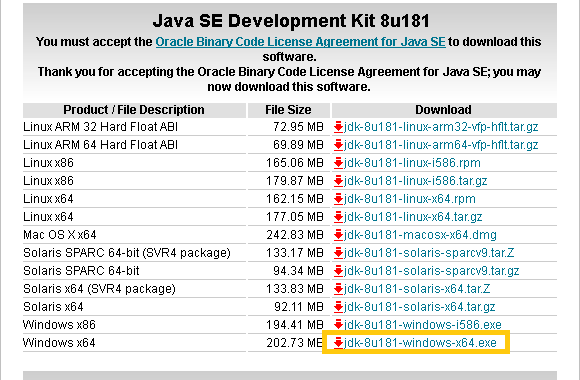
**3.1.1 Java**

There are several steps when methodology is considered. As the first step the necessary software's were downloaded. For the development process Java Development Kit was downloaded from Oracles Java site.

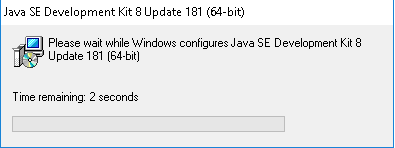
Click “Accept License Agreement.” Before clicking the download link, must accept the license agreement. The option is just beneath the JDK version number.

****

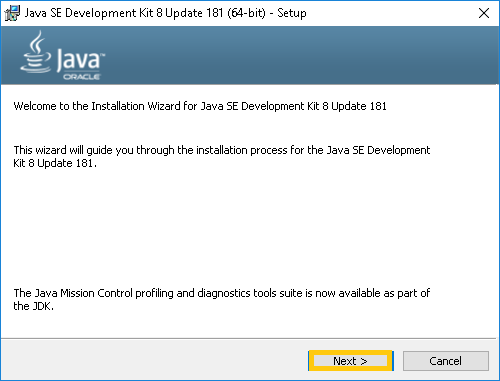
**Figure 7:JDK Accept License agreement to download**

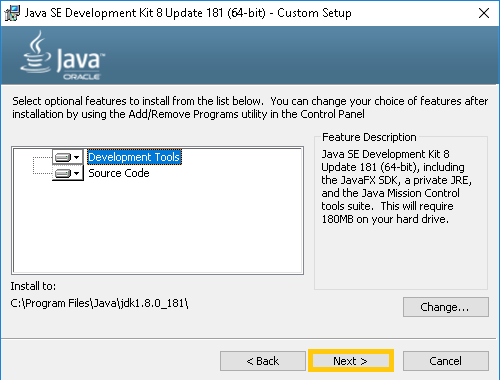
Click the download link next to your operating system. Once you click the link, follow the prompts to select a save location on your computer and start the download.

**Figure 8:Click the relevant supporting version for OS**

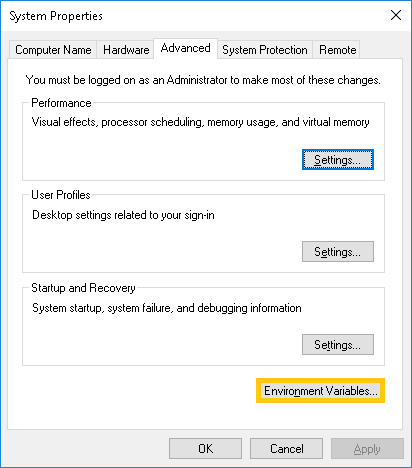
Install the JDK on your computer.

**Figure 9: JDK Install process initiation**

****Follow the installation wizard.

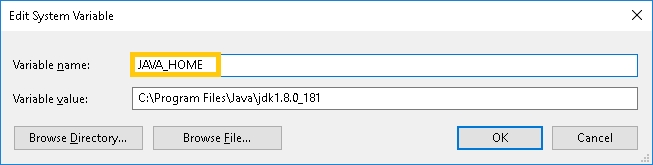
** Figure 10: Continue to install**

**Figure 11:Selecting location to install JDK**

****Right click on “This PC” and select properties and navigate to advanced system settings and find environment Variables button and Click it.

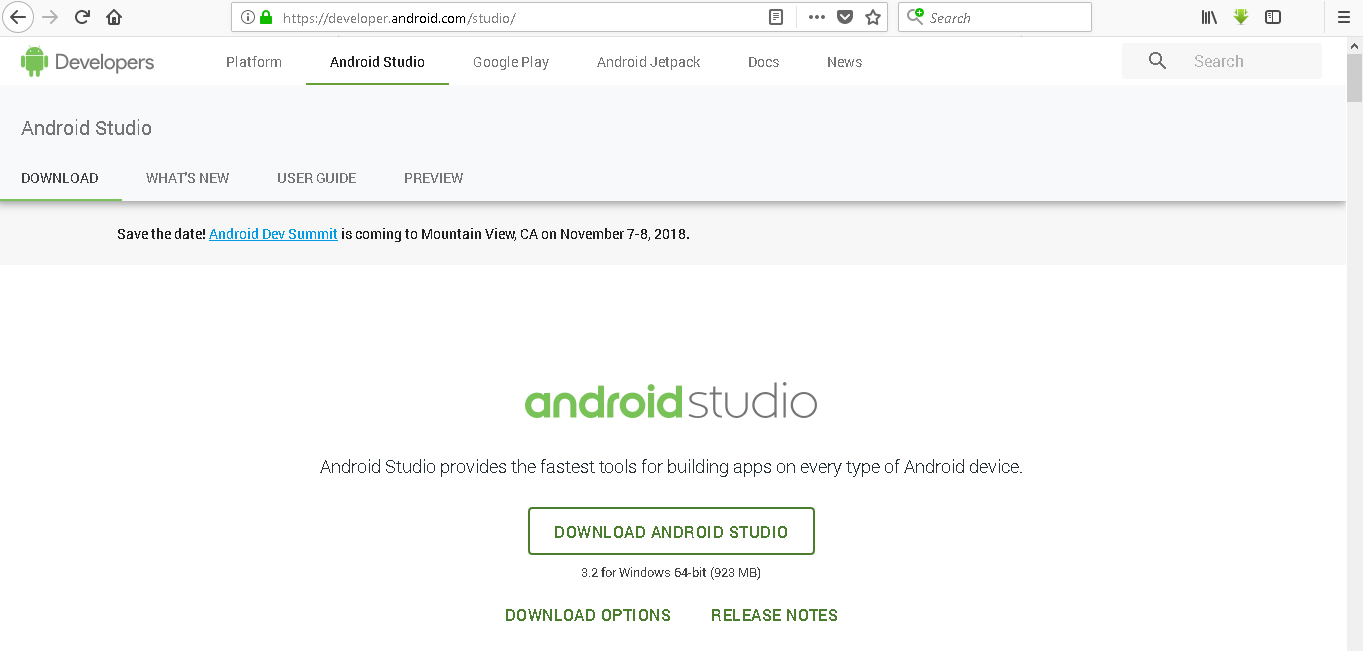
**Figure 12:Change Environment variable**

Set a System variable for JAVA\_HOME and click OK

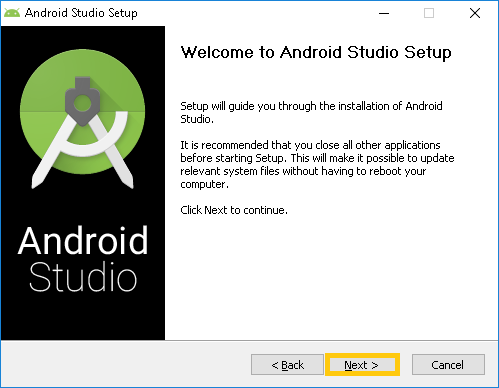
****

**Figure 13:** Edit the Java Home variable

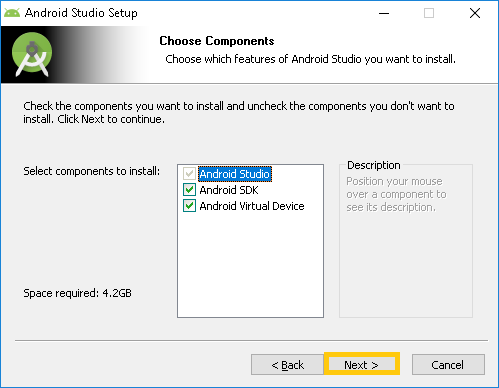
**3.1.2 Android studio**

****The latest version of Android SDK was also downloaded along with Android Studio. Android studio is a free and open source software and there are so many advantages in initiating the mobile application in android platform.

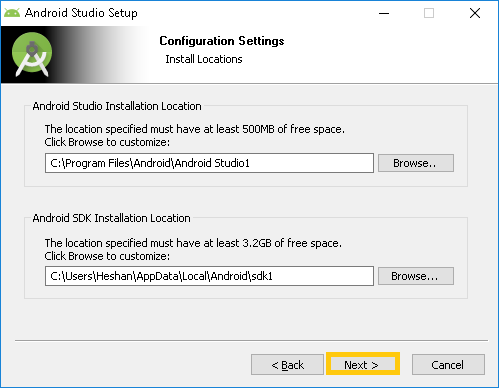
**Figure 14:Android Studio**

****The installer responded by presenting the Android Studio Setup dialog box shown in Figure 15

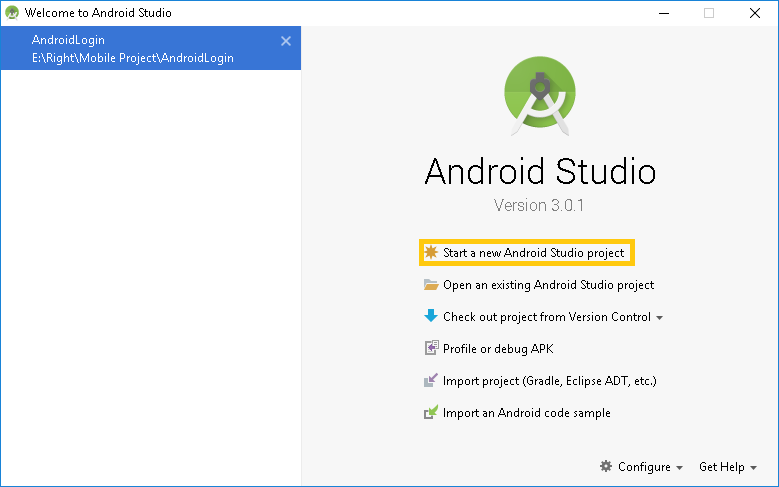
**Figure 15:Android install initial process**

****Clicking Next takes to the following dialog box, which gives you the option whether to accept or decline installing the Android SDK and Android Virtual Device.

**Figure 16:Choose Components**

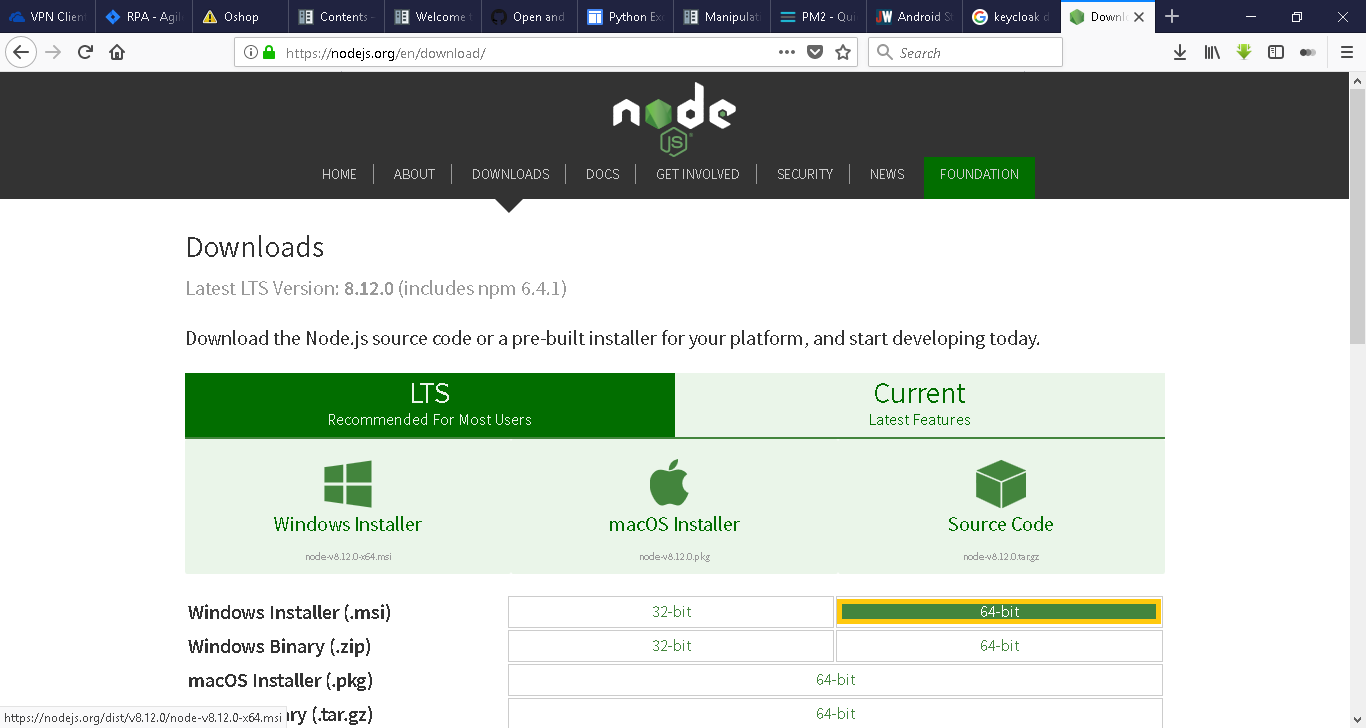
****The next dialog box invites you to change the installation locations for Android Studio and the Android SDK. Change the location or accept the default locations and click Next.

**Figure 17: Choosing a location to install**

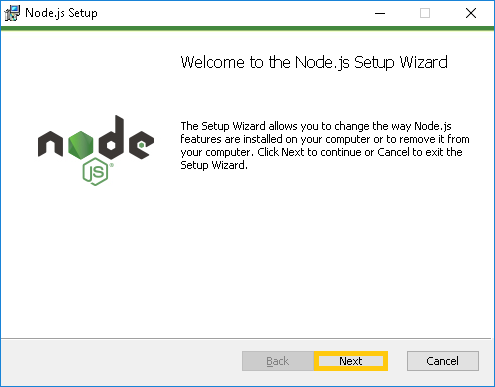
****Use this dialog to start up a new Android Studio project, work with an existing project, and more. You can access it anytime by double clicking the Android Studio shortcut on your desktop.

**Figure 18: Start a new Project**

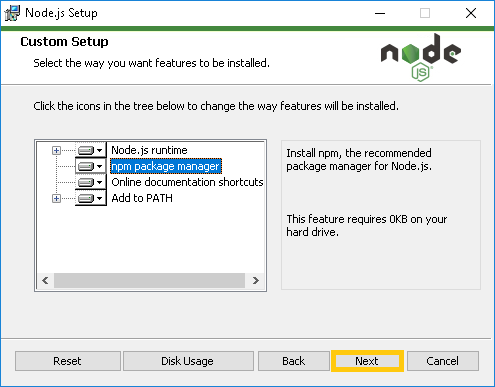
**3.1.3 Node JS**

Go to Node.js <https://nodejs.org/en/> in your computer's web browser and download the recommended version for the Operating System.

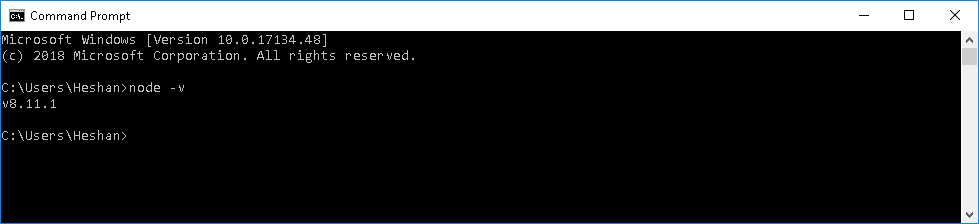
**Figure 19: Node js downloading website**

Follow the setup wizard finding a specific place to install.

**Figure 20: Node js install initiation**

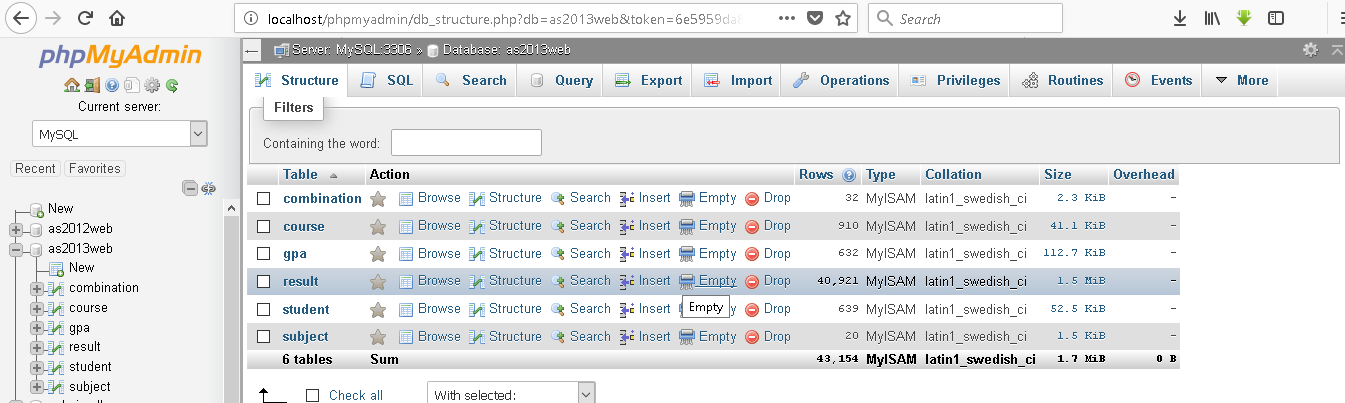
****Continue with setup clicking next

**Figure 21:Install necessary packages**

Open the command prompt and type in "node -v" and press Enter. If you get a message with a "vNumber" where "Number" is the version number of the Node.Js program you downloaded then it is successfully installed.

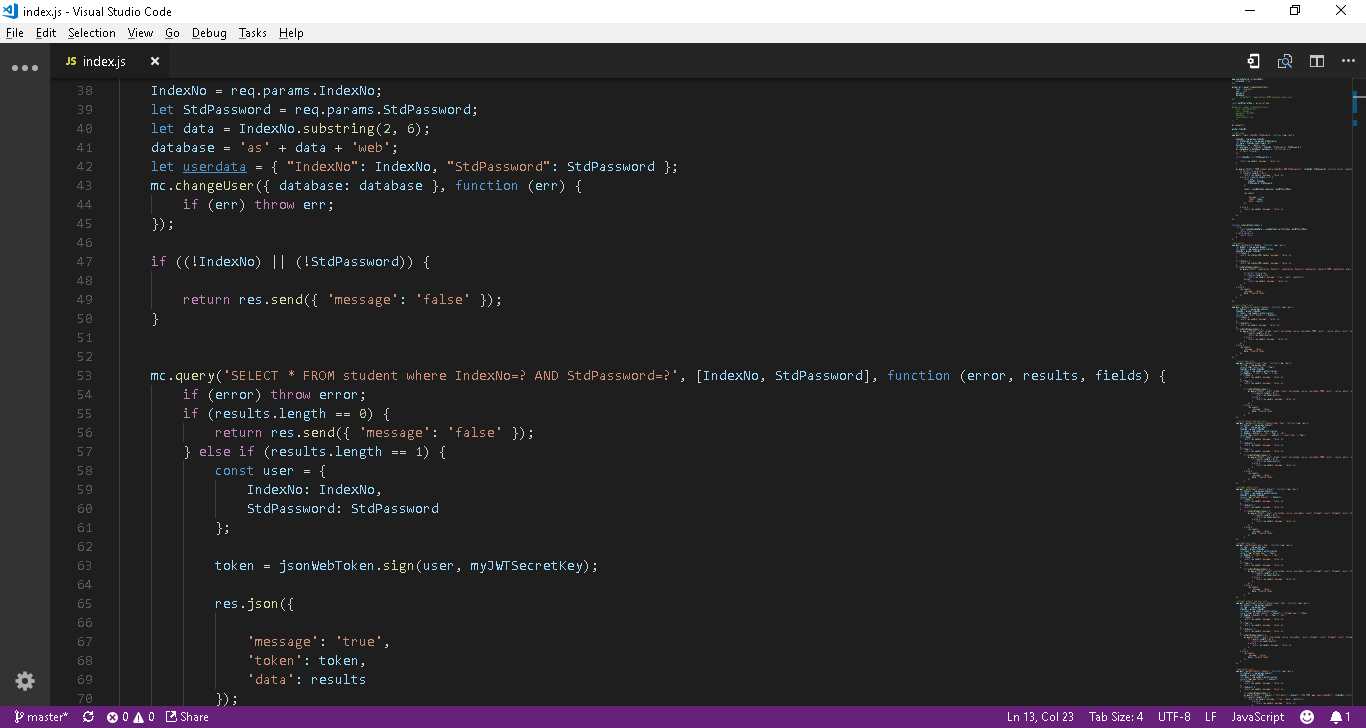
**Figure 22: Check node js version from terminal**

**3.2 Preparing the Database**

****MySQL database structure and tables were given in order to continue with the project. Each database has 6 tables to keep track of the student and results along with the subject combination

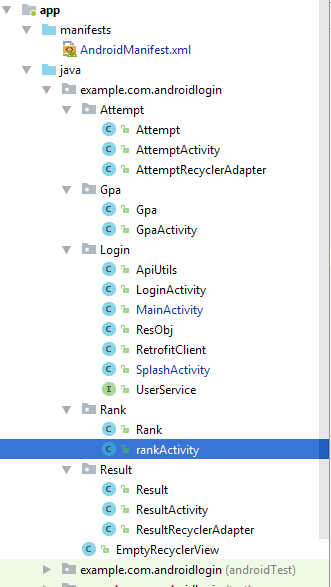
**Figure 23:MySQL database and table structure**

**3.3 Developing the API**

**** Node.js for used for querying necessary results from the database

**Figure 24: Application Programming Interface**

**3.4 Developing the Android Application**

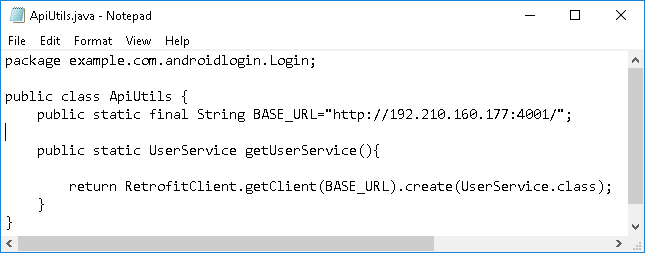
****The android application was developed with 7 activities and for the result activities Recycler Adapters was used.

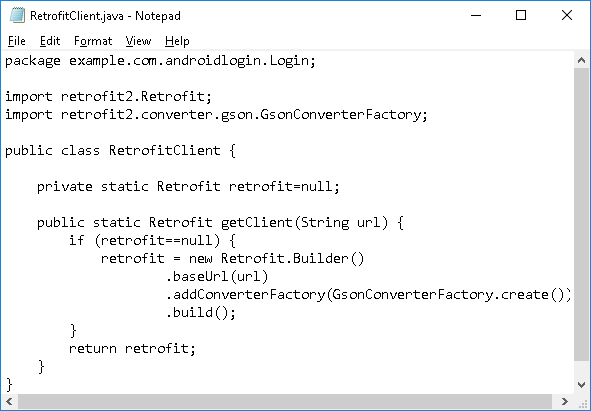
**Figure 25: Android activities**

**3.5 Retrofit**

Retrofit is a REST API Client for Java and Android. It is developed by Square Inc. and uses “OkHttp” library for HTTP Request. It is a simple library that is used for network transaction.  
It is an easy and fast library to retrieve and upload the data via Rest based web service. Retrofit mainly need three things which are following:

### 3.5.1 Retrofit Instance

You can create a Retrofit instance by ‘Retrofit.Builder()’. You have to specify the base URL and converter factory at the time of Retrofit instance.

 Figure 26: Base URL instance

**Figure 27: Retrofit Client with GSON converter**

### 3.5.2 Interface

### Retrofit turns HTTP API into a Java interface.

### Figure 28: API endpoints

### 3.5.3 Model Class

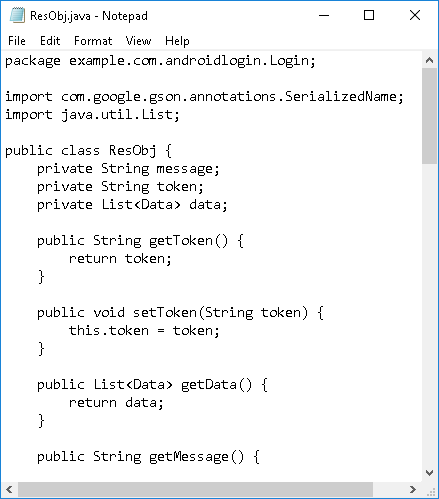
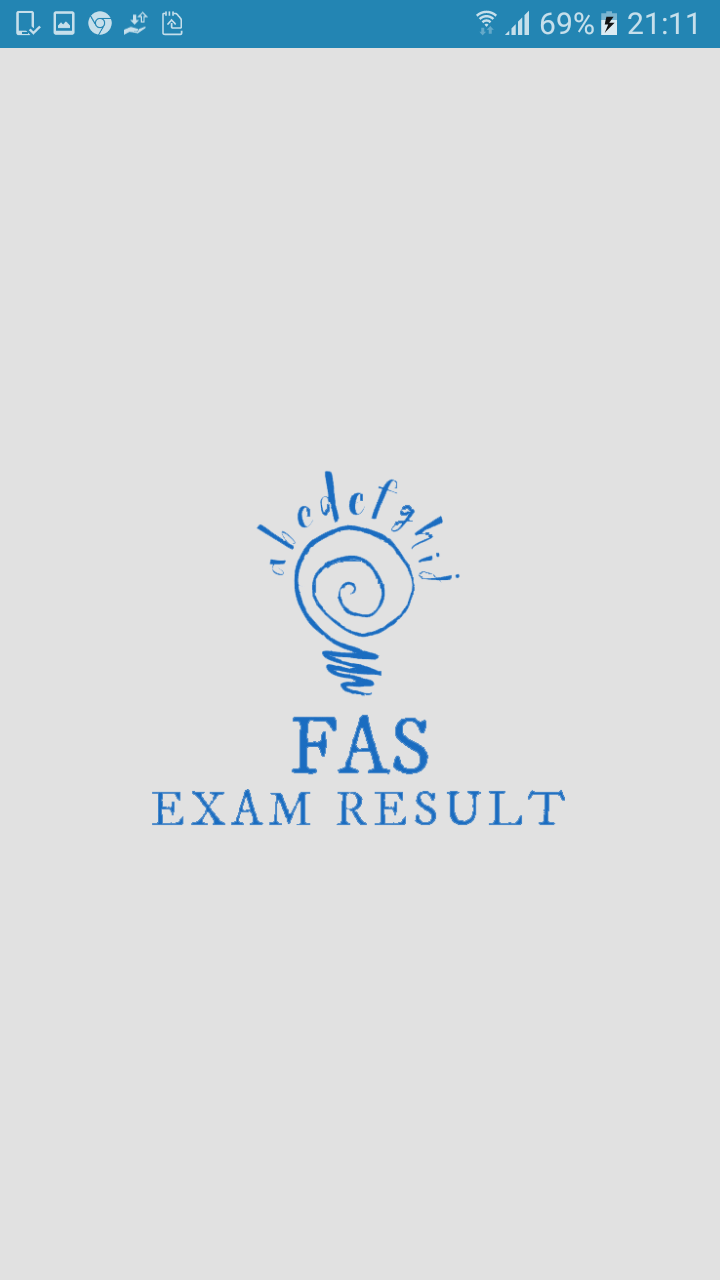
Retrofit needs a model class for sending and receiving requests using converters like GSON, Jackson.

Figure 29: Model class

Chapter 4 – Results and Discussion

**4.1 Results**

When the user clicks on the mobile app the following splash screen pops up and the app will be redirected to the login page.

**Figure 26:Splash screen**

To authenticate user need to give valid credentials, his/her Index Number as the username and NIC Number as the password. If both username and password are correct a token is generated and if the token is valid then all the API rest calls are triggered with respect to the token. If the authentication details are incorrect an error message will be displayed under username or password.

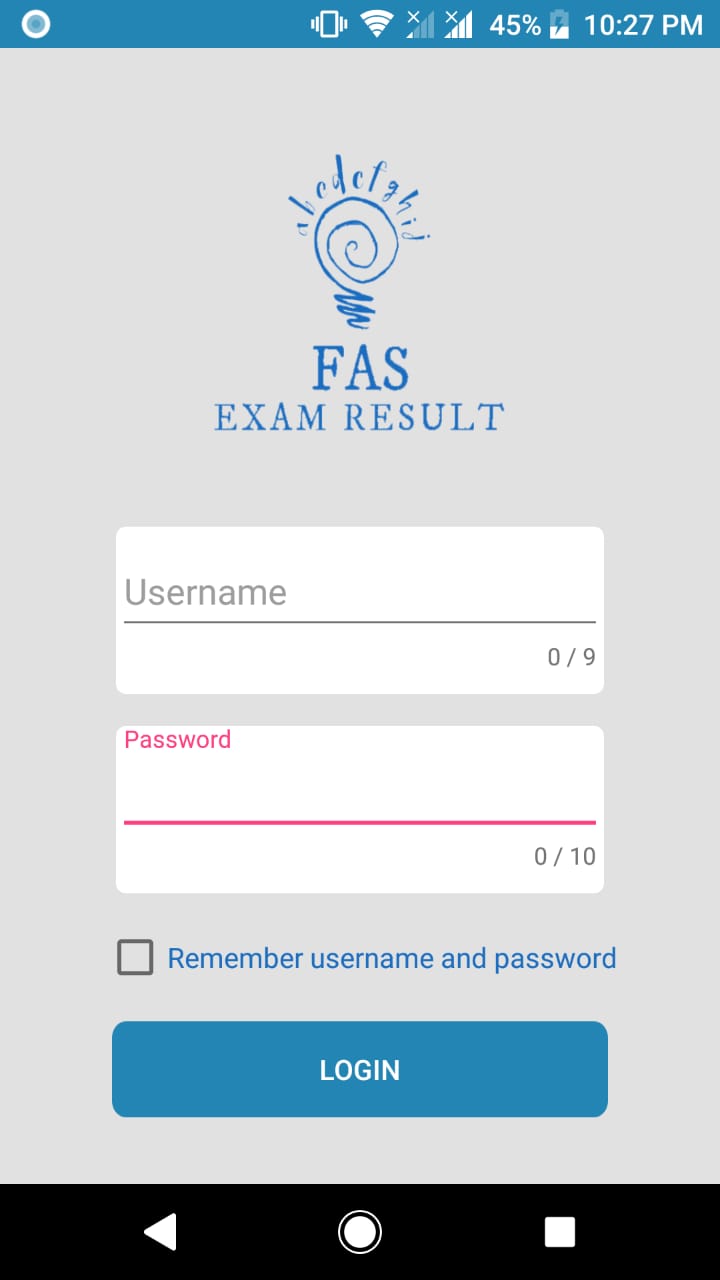
****

Figure 27:Login Activity

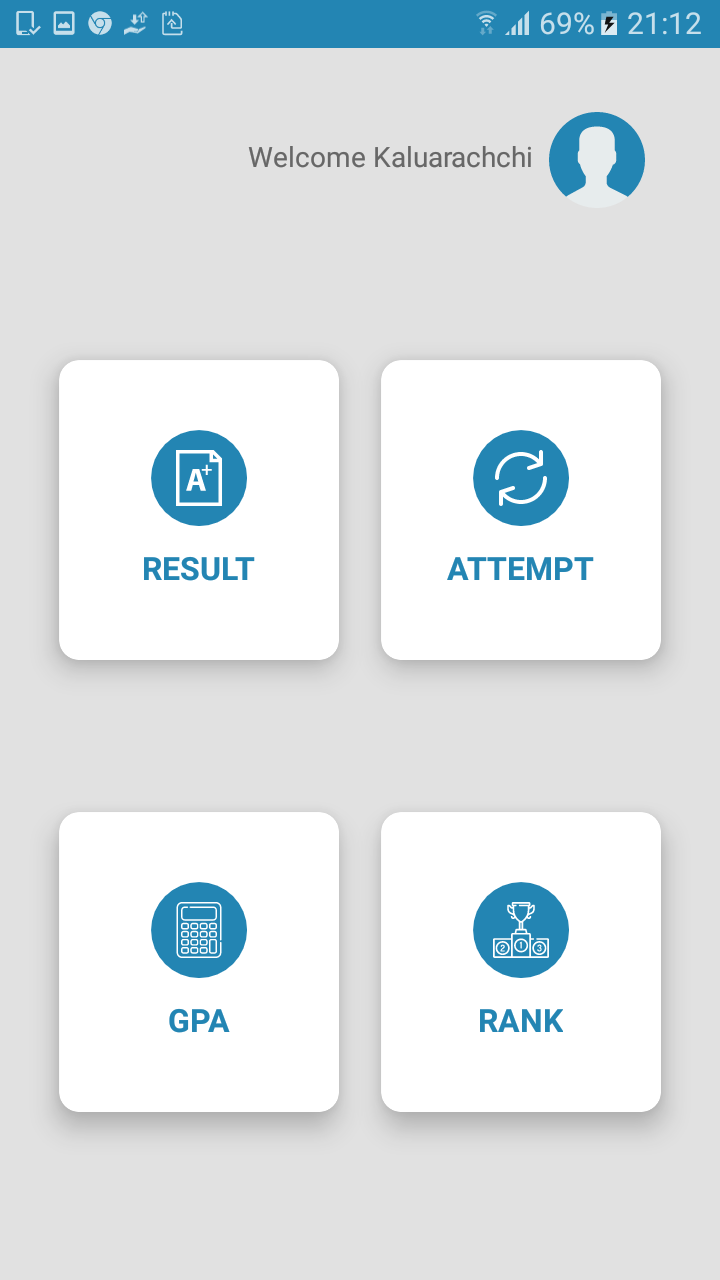
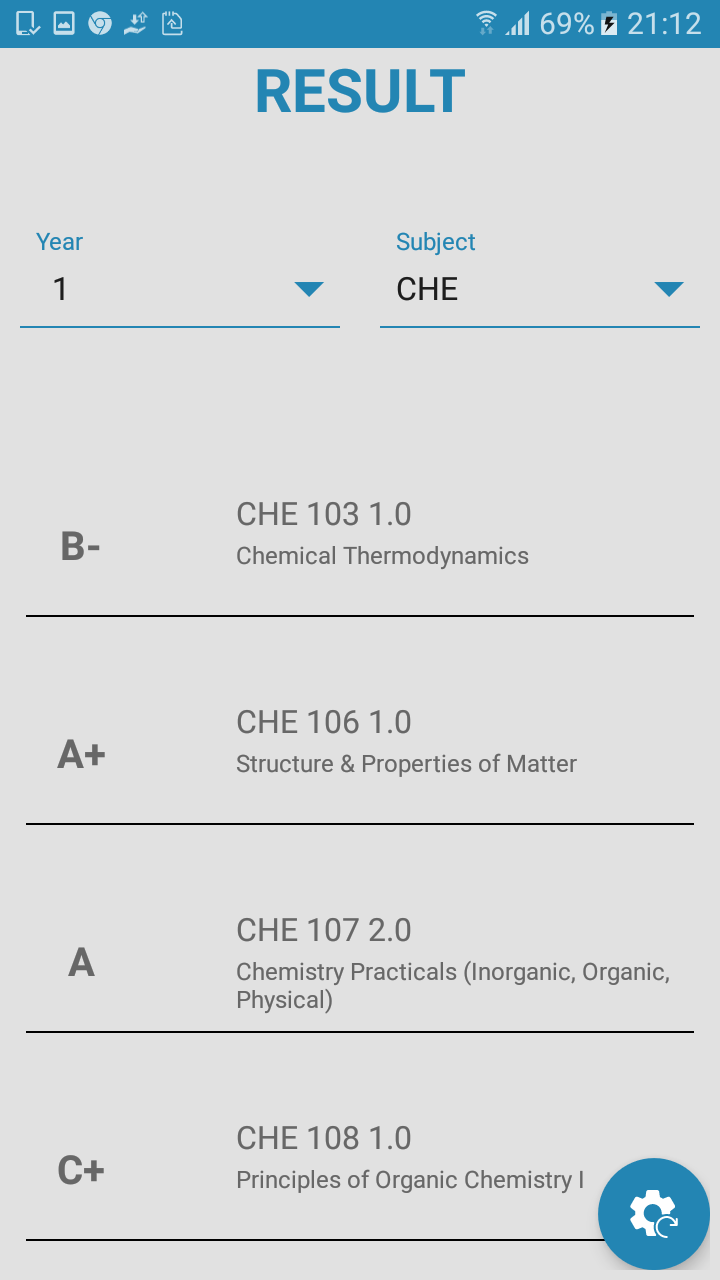
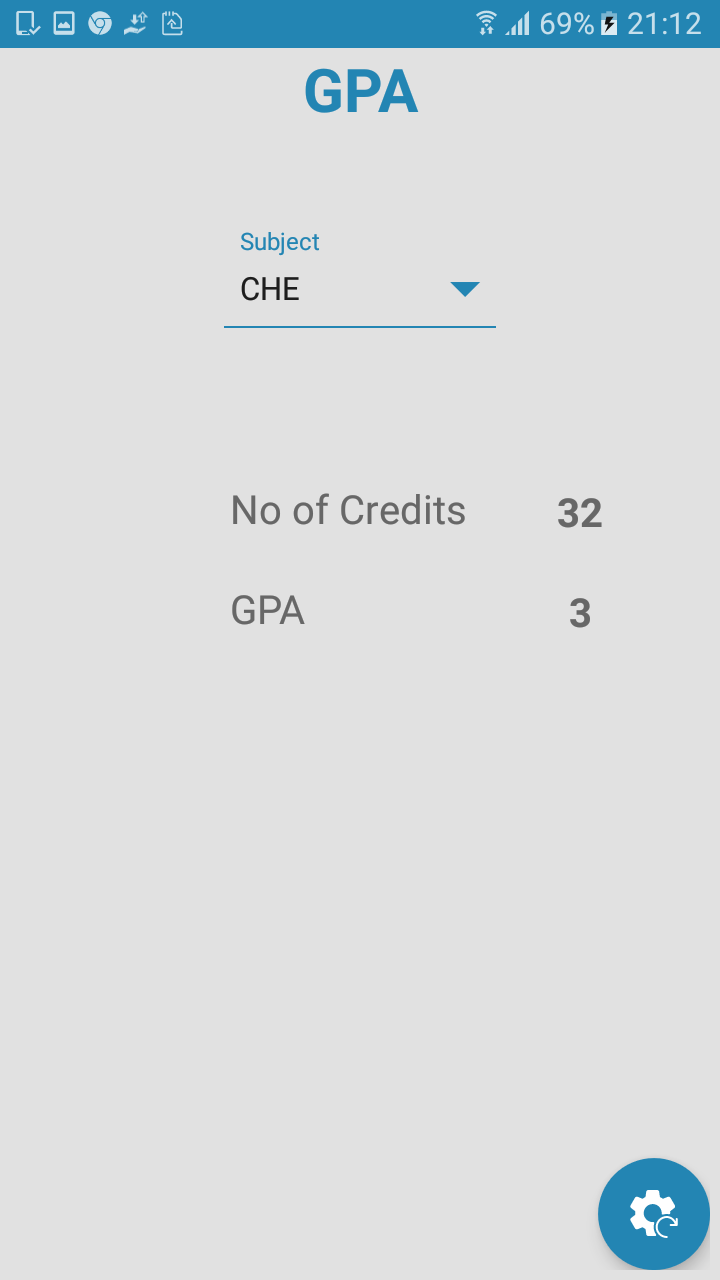
****With a successful login the user will be directed to the home page where the user can check results, attempt, GPA or rank.

Figure 28:Home Activity

****When the user navigates to the result page he/she can check results based on the year, subject or both.The user has to select the necessary option from the spinner and the result will be displayed in a recycler view. There is a floating action button in the bottom to clear the recycler view and to set the spinner to its default value. All the results are generated if the token is valid.

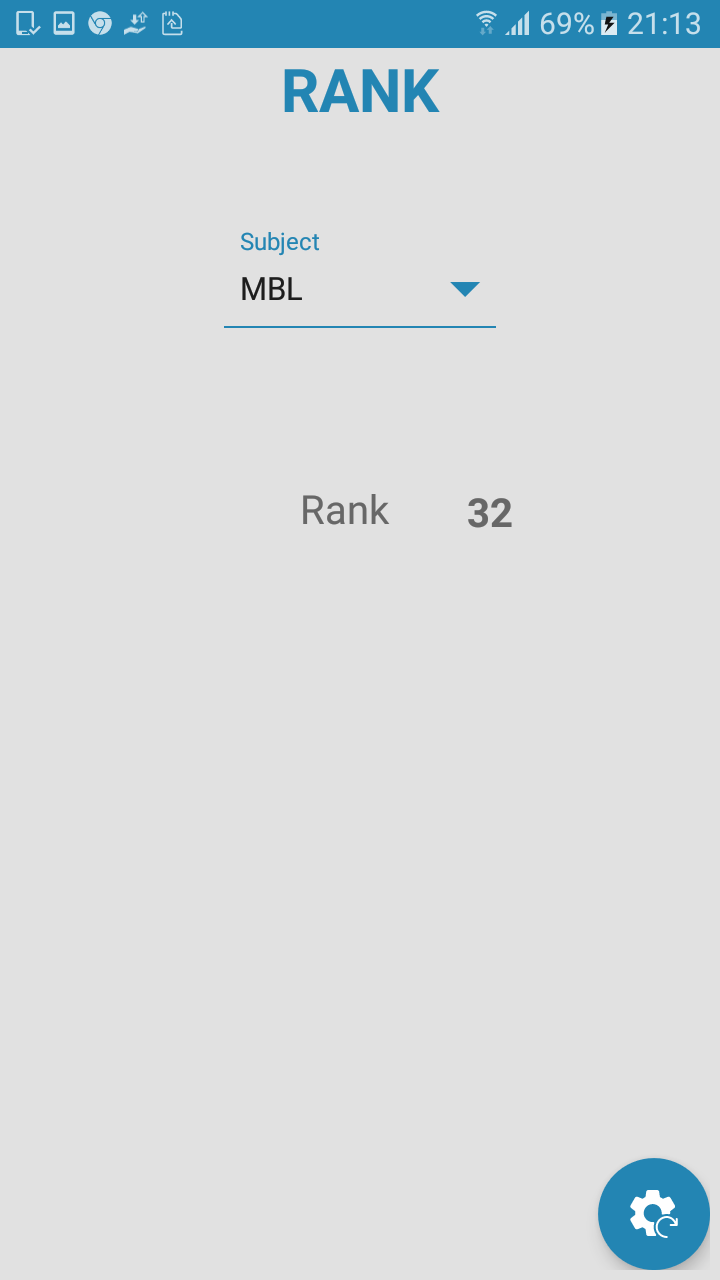
**Figure 29: Result activity after displaying results**

When the user navigates to the GPA page he/she can choose a specific subject from the spinner and the results will be displayed in a text View with respect to the number of credits the user has done. The displayed result will be cleared if the user clicks the floating action button.

****

**Figure 30:GPA Activity**

This is an added feature to the mobile app where the user can check his/her rank, subject wise by selecting from the spinner. If the user clicks the floating action button the results will be cleared and the spinner will set to the default status.

** Figure 31:Rank Activity**

**4.2 Discussion**

The primary focus of this project was to build an Android based mobile application for the students of Faculty of Applied Sciences, University of Sri Jayewardenepura. The mobile app will be an efficient way to check result than using the old fashioned web portal. In the first phase all the requirements that necessary to implement the application were gathered. Android Studio was used as the IDE to program, MySQL to handle databases and to query results, Node.js scripts as the API and Postman to check http requests/responses. In the exploding mobile technology sector, a mobile presence is necessary for any application.

In the second phase User Interfaces were designed according to the requirements and while designing developing was also started parallel. At the end of each activity the program was tested for bugs and issues. If any issues raised they were fixed at that moment and continue with the flow.

Chapter 5 - Conclusion and Future Works

**5.1 Conclusion**

This is the first mobile application implemented using android to check results of the students in the Faculty of Applied Sciences, University of Sri Jayewardenepura. The mobile application will give the users to access their results from anywhere anytime. This will affect the usage of web portals to a minimum number. Only the students in the applied Sciences is capable to use the mobile application now. So this will become an essential application for the students in the future.

**5.2 Future Works**

IOS based mobile application to check result.

Extend the application so that the other faculties can also use.

References

**[1] Android**

[**https://en.wikipedia.org/wiki/Android\_(operating\_system)**](https://en.wikipedia.org/wiki/Android_(operating_system))

**[2]Android Version**

[**https://en.wikipedia.org/wiki/Android\_version\_history**](https://en.wikipedia.org/wiki/Android_version_history)

**[3]Manifest file**

[**https://developer.android.com/guide/topics/manifest/manifest-intro**](https://developer.android.com/guide/topics/manifest/manifest-intro)

**[4]Node JS**

[**https://en.wikipedia.org/wiki/Node.js**](https://en.wikipedia.org/wiki/Node.js)

**[5]Postman**

[**https://www.getpostman.com/docs/v6/postman/sending\_api\_requests/requests**](https://www.getpostman.com/docs/v6/postman/sending_api_requests/requests)

**[6]Restful Web Services**

[**https://www.guru99.com/restful-web-services.html**](https://www.guru99.com/restful-web-services.html)

**[7]Tutorials**

[**https://www.tutorialspoint.com/android/**](https://www.tutorialspoint.com/android/)

**[8]Android app development for beginners**

[**https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBsvRxJJOzG4r4k\_zLKrnxl**](https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBsvRxJJOzG4r4k_zLKrnxl)

**[9]Retrofit**

[**https://www.youtube.com/playlist?list=PLshdtb5UWjSoFPs3AakEpH1pmSrJVxJhs**](https://www.youtube.com/playlist?list=PLshdtb5UWjSoFPs3AakEpH1pmSrJVxJhs)

A4

Left 1.25inches, all other 1 inch

Text 12pt Timesnew roman

Title 14pt Bold

Line spacing 2.0 but for figures, table captions 1.0

Numbering before introduction-roman numerals

From introduction to end-arabic

Paragrraphs should be 0.5 inches indented and justified